When records are deleted from a table through an SQL DELETE, the deleted records are referred to as "dead tuples". They are not removed from the db and the space they reside in is NOT MARKED AS BEING AVAILABLE FOR RE-USE. The same situation occurs for updated records. The VACUUM command must be run to make the "dead tuple" space available for reuse.

Running the vacuum command (without the "full" option) marks previously deleted (or updated) records as being available for re-use within the table. It does not lock the table being vacuumed. Running vacuum with the "analyze" option reads the records in the tables and generates statistics used by the queries. This information is stored in the pg_statistics table.

The SwEG's latest policy on running vacuum is to have the postgres cron submit vacuum runs for each of the standard AWIPS databases. This will prevent the possibility of multiple vacuum runs executing at the same time which can cause a slowdown in the server. A vacuum/analyze of the IHFS db has been scheduled to be submitted from the postgres cron every 4 hours.

A "vacuum full" attempts to remove deleted or updated records from the tables to make the space reusable by other tables. It physically reorders the tables. While "vacuum full" is running, an exclusive lock is placed on the table being vacuumed. This locks the table for both reads and writes. A "vacuum full" run is NOT necessary to be run on a regular basis. The SwEG is discussing the set up of an ITO Alarm for the case of a database getting very large where a run of "vacuum full" would shrink the database back to normal size.

"vacuum" and "vacuum full" can be run for an entire database or for individual tables. See Section 21.1.1 of the PostgreSQL 7.4.7 Documentation entitled "Recovering disk space" for information on strategies for running VACUUM.

Explanation of Vacuum Output

A log file is generated by each vacuum run. These logs are being written to the /data/logs/fxa directory. OHD and its PostgreSQL consultant will be monitoring these log files to watch for problems such as a slow increase in size of the db over time. We also hope to glean information from the logs which will be used to tweak the configuration parameters.

The last few lines of output from running a vacuum on an OHD db look as follows:

```
INFO: free space map: 901 relations, 6879 pages stored; 74608 total
pages needed
DETAIL: allocated FSM size: 1000 relations + 20000 pages = 178 kB
shared memory.
VACUUM
```

On the "INFO" line, "901 relations" signifies that a total of 901 tables currently exist across all databases on the server.

In the "DETAIL" line, FSM is an acronym for Free Space Map. This line shows that space has been allocated for a maximum of 1000 tables and 20000 pages for all postgresql databases on the server.